Fighting the Physics of Human Nature: Acceleration Mental Model Effects on Project Schedule and Cost Performance

Paul Newton, plnwtn@gmail.com, 206-419-4397

Abstract – Significant project and program schedule and cost overruns occur too frequently. This paper uses Systems Thinking aided by Simulation (STim) to explore how our typical mental models and mindsets can lead us to accelerate projects to the degree that, paradoxically, the program finishes later and costs more than it would have in the absence of the acceleration. It explains a real-world "quality" feedback loop that our typical mental models often discount or ignore. If this quality loop were more often and deeply embedded within our mental models we would be more cautious when planning to accelerate projects and programs. Computer simulation contrasts anticipated project schedule and cost performance dynamics arising from decisions based on both typical and improved mental models. This project acceleration insight arises from one extreme of the continuum of STim practice. Program management examples illustrate the other end of the continuum. The STim process is described, including how the STim process not only tests policies, but also provides an explicit approach for identifying policies that can improve system performance.

I. Introduction

In early 2016 the author asked participants in a senior managers' workshop to write down a problem or opportunity that mattered to them, and then to draw a diagram depicting the problem as a *pattern-over-time*. One participant sketched Figure 1 and the other participants concurred that these dynamics too frequently occur on projects. In Figure 1, the original project or project schedule is depicted as the black line labeled "original planned performance." In this example, a project acceleration decision creates the blue line labeled "accelerated planned performance"[1] that finishes earlier than the original plan. Management decides to have the project resources work as hard and fast as they can to meet the accelerated finish date as depicted by the green line in Figure 1 labeled "hoped for". Management gets as many engineers as they can working on the project and has the engineers start on tasks early in spite of

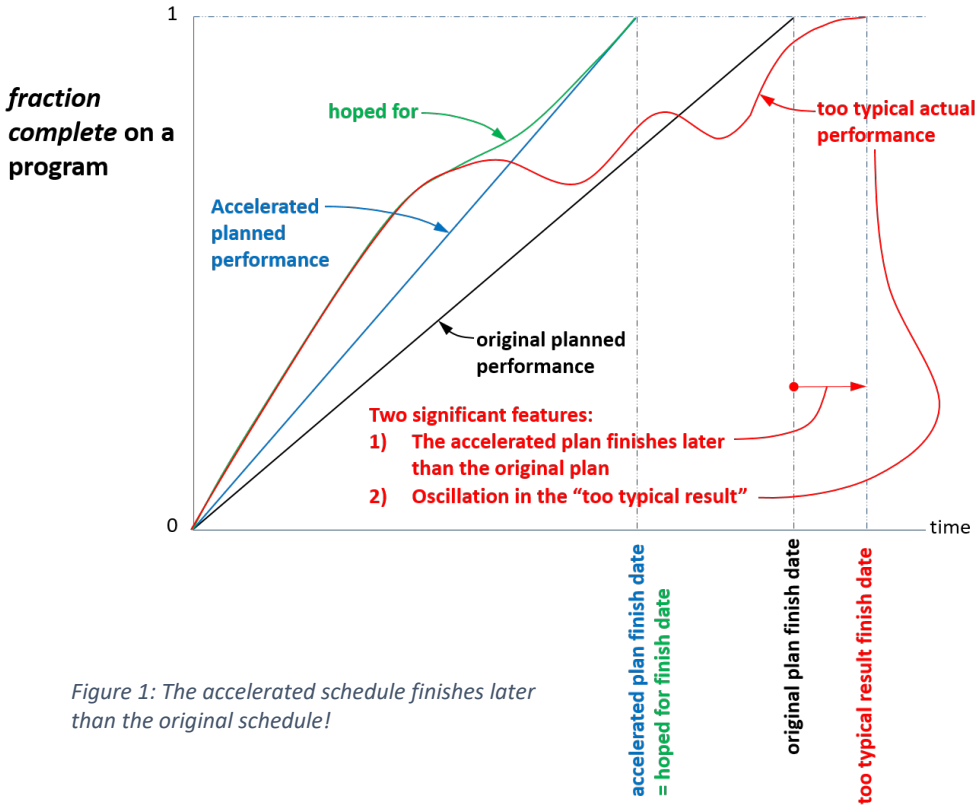


Figure 1: The accelerated schedule finishes later than the original schedule!

[1] The solid black and blue lines in Figure 1 are shown straight; however, in actuality, they would not be straight, but would probably be S-shaped.

requirements and pre-requisite tasks being incomplete. Management knows that the project is creating rework, but assumes that any rework being created can be found and corrected before the accelerated plan finish date. Indeed, the flatter part of the green line reflects the execution of anticipated rework. However, the "too typical actual performance" (the red line) is what usually happens. That is, the work ends up finishing later than even the originally planned finish date. These *patterns-over-time* recur over and over on multiple projects. *Could the attempt at acceleration itself be a cause of the project's finishing later than originally planned?*[2]

Systems thinking[3] aided by Simulation (STim) enables us to discover plausible answers to such questions. Useful here is the systems thinking iceberg shown in Figure 2. From the project acceleration story, examples of events in Figure 2 are the decision to accelerate the project, completion of tasks, and completion of the project. From a broader perspective, events could be repeated cost and schedule overruns project after project. Figure 1 shows *patterns-over-time*, the second layer in Figure 2. An answer to the question asked in the last sentence of the previous paragraph may be that our current mental model of the project system structure (the *Mental Models* layer in Figure 2) differs too much from the real project system structure (*System Structure* in Figure 2). We make our decisions on the basis of our *mental model* of *system structure*, and if our mental model is unrealistic – that is, it does not adequately match the real *system structure* – then our decisions, acting through the real *system structure*, can cause unanticipated and potentially undesirable events and patterns over time. And our *mental models* of *system structure* are significantly influenced by our *mindsets* (the bottom layer in Figure 2). Examples of mindsets are given in Part II-A of this paper.

Figure 2: The Systems Thinking iceberg

This paper explores the structure of two simple mental models of the system relevant to project acceleration decisions, *patterns-over-time* that these two mental models can create, the STim process that led to these understandings, and use of the STim process in Performance Consulting practice.

In the next section (II) we describe:

a) A hypothesis for a typical mental model underlying many project acceleration decisions, mindsets related to that mental model, and computer simulation output illustrating project schedule and cost performance anticipated by acceleration decision makers who hold that typical mental model.

b) A feedback loop often ignored by the mental models of those making project acceleration decisions, and computer simulation output illustrating more likely project schedule and cost performance.

c) Insights and recommendations.

Section III provides an overview of the continuum of STim practice, with project management examples. Section IV provides an overview of the STim process. Appendices provide the equations and replication notes for the simulations in Section II.

II. Mental models of the system underlying project acceleration decisions

Before proceeding, it is important to establish, for the purposes of this paper, the meanings of "model" and "mental model."

Model: "*an abstraction or simplification of a system*"

Mental model: "*a model that is constructed & simulated within a conscious mind*." ([6] and [7])

A. Typical mental model underlying many project acceleration decisions, mindsets supporting that mental model, and simulations of this mental model.

Figure 3 diagrams a mental model we hypothesize to underlie many project acceleration decisions. It represents a decision maker's understanding of how the system works when she accelerates the project by making the *scheduled completion date* earlier than originally planned, thus reducing the *scheduled remaining duration*. The basic idea is

[2] See the last paragraph of Section IV in this paper for further discussion of why the red line in Figure 1 oscillates.

[3] Systems thinking is discussed at length in [13], where it is advocated as the key fifth discipline of the five disciplines of the Learning Organization.

that, in response to the acceleration, when people compare their *desired remaining duration* with the now reduced *scheduled remaining duration*, they feel *schedule pressure* and increase their *productivity*. Increased *productivity* drives a faster *task completion rate*, increasing *Work Done* and *fraction complete* and decreasing *desired remaining duration* relative to what their values would have been in the absence of having accelerated the project. The decreased *desired remaining duration* better aligns with the decreased *scheduled remaining duration* from the acceleration, thus reducing *schedule pressure*. The loop continues to operate until the *desired remaining duration* equals the *scheduled remaining duration*, and voila, the project meets the accelerated *scheduled completion date*.[4]

Our hypothesis is that, if decision makers who accelerate project completion dates were to work hard at describing the workings of the system that responds to their acceleration decision, then they would eventually arrive at some version of this feedback loop. That is, the "model that they construct and simulate in their conscious mind" (see definition of 'mental model' above) would look something like this feedback loop.

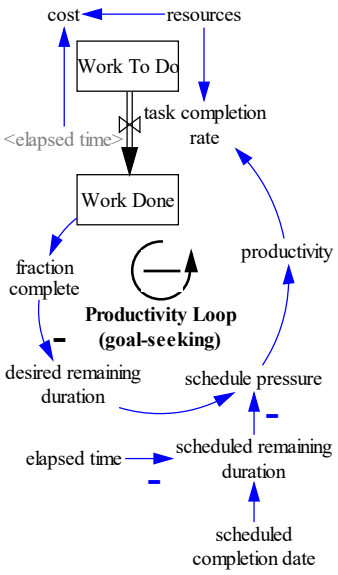
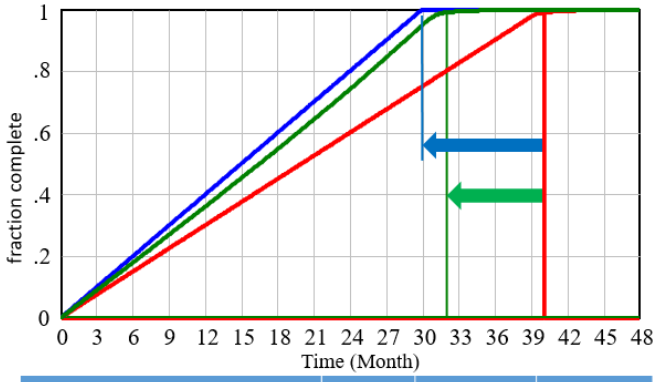


Figure 3: Mental model hypothesized to underlie many project acceleration decisions

Mindsets (the fifth layer of the systems thinking iceberg in Figure 2) that typically support this mental model might include:

- "We always produce worst-case schedules, so there is always slack in the schedule."
- "Tasks take up the time allocated to them."
- "People are lazy; they will goof off if not pushed."
- "I know we didn't finish on time on the last project, but we'll do it right this time."
- "We just have to work harder."

Figure 4 shows computer simulation results that provide an example of the nature of the schedule and cost performance that would be anticipated by acceleration decision makers who hold the typical mental model shown in Figure 3. The red lines in Figure 4 show the original feasible plan – a 40.16 month project duration that costs $12M. Because she has the mental model shown in Figure 3, the decision maker decides to set a stretch goal to finish the project in 30 months (the blue line in Figure 4), at a cost of $9M. Even though the decision maker accelerates the project to such a degree, she is likely not 100% confident that her accelerated schedule and cost will be fully met. This is because she believes that the workforce's *productivity* can respond only so much to *schedule pressure*; *productivity* will reach a limit. Not knowing what that limit is, she continues to push for the 30 month finish date, accepting risk that the project might reach a *productivity* limit and therefore not make the 30 month goal. Because of the *schedule pressure* her acceleration decision creates, she believes the project will almost certainly finish a lot sooner and cost a lot less than it would have had she not accelerated it. In Figure 4, the green finish date of 32 months and $9.6M cost are from a computer simulation of the Figure 3 mental model; this



	Original	Accelerated	Simulated
Schedule Performance (months)	40.16	30	32.06
Cost Performance (millions of $s)	12	9	9.619

Figure 4: Schedule and cost performance anticipated by acceleration decision makers who hold the typical mental model shown in Figure 3 and described in Section II-A

[4] The feedback loop in Figure 3 is a goal-seeking loop where the goal is the value of the *scheduled remaining duration* variable. For example, the loop continually acts to cause *desired remaining duration* to become equal to *scheduled remaining duration*. The "-" sign at the center of the loop signifies that the loop is a goal-seeking (negative) loop. The minus signs beside some of the arrowheads mean that the variable at the arrowhead responds to the variable at the tail of the arrow by moving in a direction opposite to the variable at the tail of the arrow. No sign at an arrowhead means that the variable at the arrowhead responds by moving in the same direction as the variable at the tail of the arrow.

simulated outcome would seem reasonable to her, and to her mind, would justify her decision to push for a 30 month finish date.

Of course, since we have a computer simulation of the mental model in Figure 3, we can use that simulation model to compare simulated project duration outcomes from a range of degrees of project acceleration. Figure 5 is a plot of simulated project durations (y-axis) that vary in response to a range of *scheduled completion dates* entered into the simulation (x-axis); each dot on the graph represents a different run of the simulation model. Note the more the project is accelerated (moving left on the x-axis), the less impact the acceleration has on reducing the simulated project duration (y-axis). Recall that the more the project is accelerated, the more *schedule pressure* is felt by the workforce; and, the more *schedule pressure* the workforce feels, the greater the *productivity* of the workforce, as previously recounted in the description for Figure 3. However, *productivity* can increase only so much, and at some point it cannot increase any further – it reaches a limit. This is why the curve becomes more and more horizontal as one moves left in Figure 5. This would not surprise the acceleration decision maker; she would expect this sort of response because she correctly believes that productivity has a limit beyond which it can increase no further. Indeed her awareness of this limit is why, as previously stated, she does not really expect that her accelerated schedule and cost will be fully met.

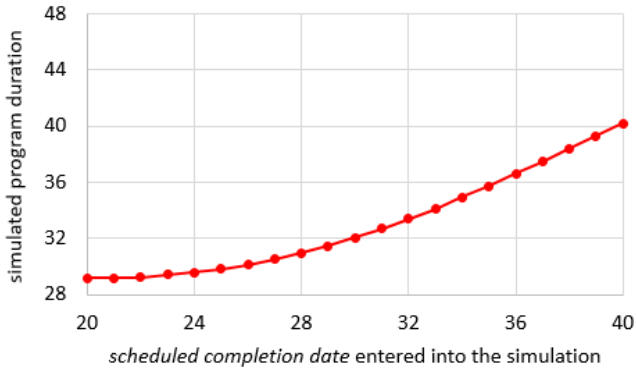


Figure 5: The earlier the scheduled completion date (moving left on the x-axis), the less the impact on reducing the simulated project duration (y-axis).

B. Improved mental model that adds a feedback loop often ignored by the mental models of those making project acceleration decisions, and simulations of the more realistic two-loop mental model.

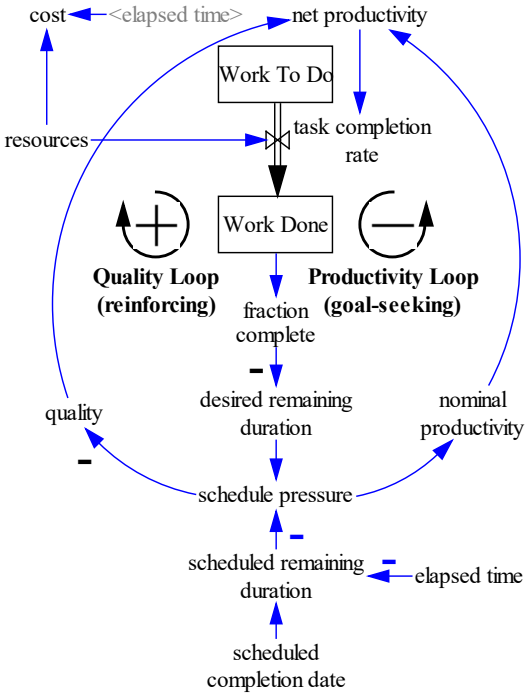


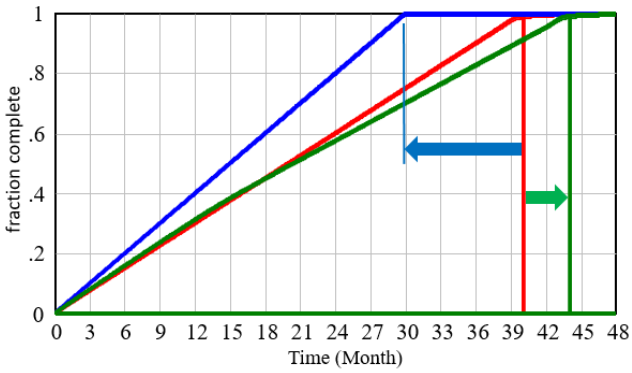
Figure 6: Adding a Quality Loop to the Figure 3 mental model. The Quality Loop is often ignored in the mental models of decision makers who accelerate project schedules.

Figure 6 expands the typical mental model from Figure 3 to include a *Quality Loop* that the typical mental model too often ignores. We have seen that when *schedule pressure* is increased (by, say, an acceleration of the project), *productivity* increases. Likewise *quality* decreases, and herein lie the origins of the *Quality Loop*. This loop's heritage is further discussed in the description of Figure 16 in [9][6] as further discussed in Section III below. Now, for an explanation of the *Quality Loop*.

As described in Section II-A when project acceleration decisions reduce the *scheduled remaining duration* to less than the *desired remaining duration*, then people on the project feel *schedule pressure*. Now, moving around the *Quality Loop*, increased *schedule pressure* causes people to take shortcuts and work sustained overtime until fatigued, both of which reduce work *quality*. *Quality* varies from 1 (perfect work) down to 0 (completely imperfect work). Less work *quality* means less *net productivity* (*net productivity* = *nominal productivity* * *quality*). Less *net productivity* reduces *task completion rate*, meaning that at any point in time there is less *Work Done*, a reduced *fraction*

complete, a longer *desired remaining duration*, and more *schedule pressure* than there would have been had the project's *scheduled completion date* not been accelerated.[5]

Although managers may acknowledge the existence of the *Quality Loop*, it may be that our project acceleration decisions do not take this loop into account. For telling cross-industry information on both cost and schedule overruns on large projects, differentiated by design and build phases, see Figure 16 in [9] and its accompanying description.[6] Although it may be that not all of these overruns are caused by over-acceleration, our suspicion is that a significant number of project overruns are at least partially caused by acceleration decisions where mental models at least discount, if not ignore, the quality loop in Figure 6.



	Original	Accelerated	Simulated
Schedule Performance (months)	40.16	30	44.09
Cost Performance (millions of $s)	12	9	13.23

Figure 7: Schedule and cost performance anticipated by acceleration decision makers holding the more realistic mental model shown in Figure 6 and described in Section II-A. Compare to Figure 4.

Before continuing, refer back to Figure 4 to see schedule and cost performance that would not surprise decision makers who, on a project with a feasible 40 month schedule and $12M cost, accelerate the project to 30 months, and then observe project completion in 32 months at a cost of $9.6M. Recall that this is based on the mental model in Figure 3.

Figure 7 shows simulated completion date and cost results that would be anticipated by someone who holds the more plausible mental model in Figure 6. The same 40 month and $12M project (accelerated to 30 months) actually finishes in 44 months at a cost of $13.23M. Given that this is the more plausible project performance because it contains the quality loop in Figure 6, it is likely that the decision maker in the previous paragraph would actually experience results more like Figure 6 rather than her anticipated Figure 4 results.

Again, because we have a computer simulation model of the more plausible mental model in Figure 6, we can use that simulation model to compare simulated project duration outcomes from a range of degrees of project acceleration. Figure 8 repeats the graph from Figure 5 (red) and adds a graph (green) of the outcomes produced by the system in Figure 6; recall that each dot on the graph in Figure 8 represents a different run of the simulation model and that the green output in Figure 8 reflects the combination of both the *Productivity* and *Quality* loops in Figure 6, whereas the red output in Figure 8 reflects only the *Productivity Loop*.

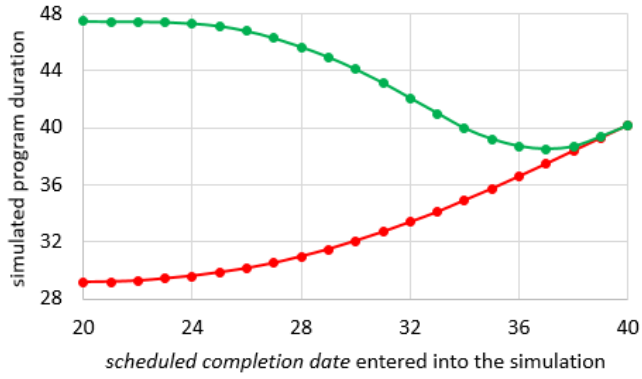


Figure 8: Simulated project duration responses to multiple acceleration durations for both the system in Figure 3 (red - a repeat of Figure 5) and the system in Figure 6 (green)

Note that with the addition of the *Quality Loop* (the green simulations in Figure 8), some acceleration reduces overall project duration, but more acceleration causes greater project duration. The technical reason for this is different responses of the *Productivity* and *Quality* feedback loops to varying degrees of *schedule pressure* caused by accelerating the schedule. In response to just a little acceleration, the *Productivity Loop* responds more strongly to the *schedule pressure* created by project acceleration than does the *Quality Loop*, thereby causing an earlier finish date than the original schedule. But with more acceleration, the *Quality Loop* responds more

[5] The *Quality Loop* in Figure 6 is a reinforcing loop. In response to a change in the value of any variable in the loop (in this case an increase in *schedule pressure* due to project acceleration), the loop acts to reinforce that change. For example, the loop continually acts to cause *desired remaining duration* to diverge from *scheduled remaining duration*, thus keeping *schedule pressure* high. The "+"sign at the center of the loop signifies that the loop is a reinforcing (positive) loop.

[6] [9] presents empirical evidence from design and build overruns in both budget and schedule for a sample of ten aerospace, shipbuilding, and civil construction projects

strongly to the *schedule pressure* created by project acceleration than does the *Productivity Loop*, resulting in a later finish date than the original schedule. And these responses represent the physics of human nature; when stressed somewhat above normal, human performance generally improves; but, when stressed well above normal, human performance generally declines. The simple model in Figure 6 represents this differential performance as being caused by the combination of two feedback loops, the performance-improving *Productivity Loop*, and the performance-reducing *Quality Loop*.

Recall that our original question at the end of the second paragraph in the introduction was, "*Could the attempt at acceleration itself be a cause of the project's finishing later than originally planned?*" Note that the green trace through the simulations in Figure 8 both answers this question in the affirmative, and, other than the oscillations in the red line in Figure 1, also confirms that we have developed a system structure that can replicate the *patterns-over-time* in Figure 1.[7]

C. Insights and recommendations

For decision makers who have the authority to accelerate schedules, the primary recommendation is to take into consideration the effects of *both* the *Productivity Loop* and the *Quality Loop* in Figure 6 when making schedule acceleration decisions. Projects will tend to respond to acceleration decisions in the manner traced by the green simulation points in Figure 8. It is therefore best to base acceleration decisions on a mental model that does not discount the *Quality Loop*.

A second recommendation for decision makers is that, when making acceleration decisions, it is important to be working from a feasible schedule. The original simulations in Figures 4, 5, 7 and 8 (a 40 month and $12M project) are feasible. That is, the base run *scheduled completion date* (40 months) equals initial *Work To Do* (720 tasks) divided by the product of *resources* (20 people), *expected productivity* (1 task/person/month), and *expected quality* (0.9 fraction). Feasible completion dates are those that are greater than or equal to the result of this calculation, whereas infeasible scheduled completion dates are less than this calculated value. In the real world, it is difficult to be certain that the initial scheduled completion date is feasible due to uncertainties for productivity, quality, and perhaps even for the number of resources and tasks in the work statement. However, every attempt should be made to ensure a feasible initial scheduled completion date. In this regard it is useful to keep records of initial and actual scope, and the actual productivity and quality of the workforce from project to project. Such information can then be used to provide a baseline for determining feasible initial scheduled completion dates for similar new projects [5]. It may be that the current scheduled completion date that the decision maker is considering to accelerate is actually already an infeasible schedule. If so, it is important that the decision maker have a sense of the difference between the current scheduled *completion date* and what that date would be for a feasible schedule. Any difference should be taken into account when making acceleration decisions.

A different set of insights and recommendations is important for managers who have a project acceleration decision imposed on them. First, keep your team's work *quality* as high as possible by protecting your team from feeling too much *schedule pressure*. Second, sense the team you manage; try to find ways to measure the *productivity* and *quality* of their work, and to measure the effects that different degrees of *schedule pressure* have on their *productivity* and *quality*. Of course, measuring such attributes of a single individual, much less an entire team or workforce, is difficult, but explicitly working to understand these attributes will improve management practice. Finally, understanding the mental models and the *patterns-over-time* described and illustrated in this paper provides context for the importance of striving to sense your team.

One last recommendation is appropriate for all readers of this paper. Note that two types of *productivity* are named in Figure 6, *nominal productivity* and *net productivity*. We talk a lot about *productivity*, but rarely do we make this important distinction. *Nominal productivity* is productivity prior to the effects of *quality* on *productivity*, whereas *net productivity* equals *nominal productivity* multiplied by *quality*. That is, *net productivity* includes the effects of *quality* on productivity.

Sections III and IV of this paper independently build on Sections I and II. Section III describes the continuum of STim practice. One end of the continuum is exemplified by the STim work in Sections I and II. The other end of the continuum will be illustrated by the application of STim to specific development projects. Section IV describes the STim learning process that guided the thinking underlying Sections I and II.

[7] Potential causes of these oscillations are discussed in the last paragraph of Section IV.

III. The continuum of STim practice with project management examples

The view of the system structure of projects in Figure 16 in [9][12] as well as Figures 3 and 6 herein, is very different than our usual view of the system structure of projects as expressed in Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT) models. From [9, p. 135],

"*Research, design, and development projects are notorious for failing to achieve cost and schedule budgets, in spite of considerable effort over the years toward improving project management. The failure to improve project performance results, in large part, from models which do not treat projects as the complex dynamic systems which they are. In particular, current planning approaches* [e.g. CPM/PERT] *fail to consider:*

- *Rework (especially undiscovered rework);*
- *Feedback effects, often vicious circles, which cause productivity and work quality to change over time; and*
- *The knock-on effects of performance from one project phase to another.*"

Figure 16 in [9][12] also contains explicit variables for productivity, quality, Undiscovered Rework, morale, workforce skill and experience, and other variables which traditional CPM and PERT models do not contain.

The STim process is for the most part applicable across the continuum of STim practice shown in Figure 9. The project acceleration example presented earlier in this paper is at the management insight, intuition and rules-of-thumb end of the continuum shown in Figure 9. The rest of this section of the paper describes STim simulation models at the right end of the continuum, which are much more expensive to develop, but which also have more well-defined business performance benefits. STim practice across the full range of the continuum is useful in Performance Consulting practice.

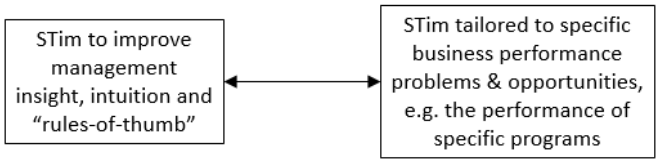


Figure 9: Extremes of a continuum of STim practice

Simulation models of projects based on the system structure view in Figure 16 of [9][12] could be developed and used both in planning and execution on every major project. Were this to be done, then future similar projects could be better planned and executed by strategically adapting the model of the previous project project to the new project [5]. Pages 242-244 of [5][8] explains how such project dynamics models can be used:

***"Pre-Project*:**

"*Bid or Plan Analysis: The model is used to establish and/or test the feasibility of schedule and budget given scope and other strategic requirements. Ideally, a model of an ancestor program is first used to determine the characteristics of a typical project in the organization, including normal productivity, rework, management practices, etc. The model is then adapted to the scope and anticipated external conditions of the proposed project, and used to assess cost/schedule tradeoffs for the proposed project. If a bid has already been submitted, the model is used to assess the assumptions required to make the bid, e.g., productivity, rework, external conditions, and actions to bring the project in as close as possible.*"

"*Competitor Analysis: Publicly available information is used in conjunction with the simulation model to estimate what the program might cost a competitor, and therefore provide a range of possible competitive bids.*

"*Risk Analysis: The model is used to determine the impact of possible changes in external conditions on the performance of the project versus the bid or plan. A simulation that reflects the project plan provides a baseline against which alternatives are measured. The direct impacts of possible changes in external conditions (specification or scope changes, design difficulties, risks such as labor shortages, late vendor design or material deliveries, etc.) are input to the model (alone and in combination) and simulation results compared to the baseline. Risks of high probability and/or those producing a significant overrun for the project warrant careful monitoring and/or mitigating actions.*

[8] [5] also uses a case study from Hughes Aircraft Company to illustrate how project dynamics models can be used.

"*Mitigation Analysis: The model is used to determine changes in program schedules, interim milestones, resourcing, etc., which minimize the consequences of risks."*

"*During-the-Project*:

"Risk Management: The model is used to determine the impact of project risks that actually materialize (we call them "Issues"). *First, the baseline simulation is compared to a simulation in which the direct impacts of the risks are included. Then, the model is used to determine changes in the program that minimize the consequences of that specific risk (for example, changes in schedule duration, interim milestones, phase overlap, additional staff, new processes or methods).*

"Change Management: Change management is a subset of risk management, but where the changes (usually scope increases or design changes) are often at the request of an external customer and therefore generally involve adjustments to the contracted cost and schedule. The model is used to determine the likely full cost and schedule implications of specification and scope changes by comparing two simulations: the baseline simulation (or current simulation of project) and a simulation in which the direct impacts of the changes are included. The latter simulation determines the indirect, secondary and tertiary effects of the direct impact on the project. These results are used as the basis for negotiating reasonable compensation and/or for designing actions to mitigate the cost/schedule impact of the change.

"*Evaluation of Process Changes: The model is used to assess the total impact of process or organizational changes, such as computerized design, new tools, integrated product design, and teaming. Implementation of change is often disruptive and involves short-term costs before long-term benefits are realized. Without an analysis of these dynamics, change programs are often abandoned before they have a chance to succeed.*" [See [17] for an example.]

"*Post-Project*:

"*Benchmarking and evaluation of best practices: Without a simulation model, it is very difficult to compare the performance of projects in a meaningful way. How much of the difference results from different products? From different external conditions? From different management practices? With a model, it becomes possible to answer these questions. First, models are developed and calibrated to the different projects. Then, differences in external conditions and scope/complexity are removed. What remains can be attributed to differences in management actions. The improvements from specific actions are further assessed by changing the actions in the simulation and comparing the performance of the simulated projects.*

"*Training and Development: The training and development of future project managers can be enhanced through the use of simulation models of projects. First, models are used as ''flight simulators'' to allow practice and learning. Second, the lessons about what works are inferred from past projects, as described above, and communicated to the next generation of managers.*

"Many of these model uses were performed for the Peace Shield Program described in the remainder of the article." ([5], pp. 242-244)

Such uses are further illustrated in several case studies from the U.S. Navy, Northrup Grumman, Hughes Aircraft, Litton Industries, and Fluor Corporation [1, 4 and 5].[9] Such a model-informed planning and execution process would enable achievement of structured quantitative continuous improvement in cost and schedule performance from program to program. However, because such models contain less task and precedence detail than the classical models using CPM and PERT, they should be used in addition to, and not replace, the classical models.

One further contribution of STim to program performance is the interaction of resources across a system of programs, essentially an expansion of the system structure in [9]'s Figure 16.[12] For example, [10][10] illustrates how

[9] U.S. Navy Trident Submarine Program in [4]. Northrup-Grumman, Hughes Aircraft, and Fluor Corporation in short videos and papers in [1]. Hughes Aircraft in [5].

[10] [10] is one of its author's early papers on this topic and others, especially his Ph.D. students, have since built on his work. The STim model in this particular paper resides more at the left end of the continuum in Figure 9.

we allow our program resources to get trapped into focusing effort on late-stage programs to the detriment of early-stage programs. The simulation model in the paper enables testing of strategies for breaking this vicious cycle.

IV. Overview of the STim process

The process for doing STim steps through the five layers of the systems thinking iceberg in Figure 2. Working down the iceberg, the top level activities for each iceberg step are:

1) *Events* step: Understand the nature of the problematical events.
2) *Patterns-over-time* step: Formulate the problem using *patterns-over-time*, showing both problematical patterns and desired patterns.
3) *System Structure* step: Hypothesize a plausible system structure that could cause the *patterns-over-time*. Use computer simulation to test that the hypothesis can produce the problematical *patterns-over-time*.
4) *Mental Models* step: Hypothesize plausible mental models that could cause the problematical *patterns-over-time*. Hypothesize improved mental models that could achieve the desired *patterns-over-time*. Use computer simulation to test that both sets of hypotheses can cause their respective *patterns-over-time*.
5) *Mindsets* step: Hypothesize mindsets that either may arise from, or may cause, the mental models that could cause both the problematical and the desirable *patterns-over-time*.

The progression of work generally steps down the iceberg, but with at least occasional, and often frequent, returns to earlier steps, and then back down through the steps yet again.[11]

There are some additional important activities in some of the steps. Before leaving the *patterns-over-time* step it is useful to write down questions that are raised by the *patterns-over-time*. For example, looking at the *patterns-over-time* in Figure 1, we asked two questions:

1) Why does the accelerated plan sometimes finish later than the original plan?
2) What causes the oscillations in the "too typical actual performance" *pattern-over-time* (the red line)?

Asking these questions was important, because we decided to focus our attention in the lower iceberg steps to only the first question. Notice that there are no oscillations in Figure 4 and Figure 7, and to this point in the paper no further discussion of oscillation. Looking only at this first question, we have been able to arrive at some useful insights and recommendations. Of course, we could expand our investigation to include the oscillations. However, note that had we initially set out to address both questions, it may have been much more difficult to arrive at the useful insights and recommendations we have so far discovered. This is because the system structure that produces the oscillations may have masked the structure necessary to arrive at these insights and recommendations [12].

Likewise, there are additional activities in the *system structure* and *mental model* steps:

1) Develop a qualitative (not simulatable) hypothesis of a plausible system structure or mental model that it is believed could create the *patterns-over-time* in the iceberg's second layer.
2) Develop a plausible, simulatable hypothesis of a system structure or mental model that corresponds to the qualitative hypothesis. To do this, iteratively formulate simulatable bits of the full hypothesis, alternating with simulation of those bits to test the *patterns-over-time* they produce against plausible expectations for those *patterns-over-time*.
3) Iterate between 1) and 2) until a plausible, simulatable hypothesis exists that, when simulated, produces the patterns over time in the *patterns-over-time* step.

The reader may be wondering how to hypothesize a system structure. Since project dynamics is a prominent application area of system dynamics, one approach is to look for examples in the project dynamics literature [11]. The project dynamics literature [2] reveals a shared understanding, across many practitioners and researchers, of the basic generic system structure of projects/programs. Figure 16 in [9] is a representation of this generic project/program system structure.[12] Studying this system structure for the simplest set of feedback loops that could address the first question, "Why does the accelerated plan sometimes finish later than the original plan?", led to a hypothesis that the following two loops from Figure 16 in [9][12] could be sufficient to create these dynamics: Loop 1: productivity → progress → Work Really Done → perceived progress → expected completion time → schedule pressure → productivity; and, Loop 2: quality → progress → Work Really Done → perceived progress → expected completion time → schedule pressure → quality; with scheduled completion time → schedule pressure in both loops. STim

[11] Chapter 3 of [16] describes one of many modeling processes similar to the STim process, but which does not differentiate between mental models and mindsets.

[12] For a clear stepwise unfurling and clear explanation of the system structure, read the text in [9] that describes [9]'s Figures 9, 13, 14, 15, and 16.

computer simulations like those shown in Figures 4, 5, 7, and 8 supported our hypothesis. Note that these two loops in Figure 16 of [9][12] are equivalent to the *Productivity Loop* (Figures 3 and 6) and the *Quality Loop* (Figure 6).

A key aspect of this system structure are two nonlinear functions that were strongly hinted at in the next to last paragraph of Section II-B. These nonlinear functions are the effects of schedule pressure on productivity and quality, respectively. The functions used in the STim simulation model (Figure 11 and Figure 12 in Appendix 1) are very similar to the same functional relationships shown in many other project dynamics papers.[13]

Having answered the first question, we are now in a position to begin to investigate the second question, "What causes the oscillations in the 'too typical actual performance' *patterns-over-time* (the red line) in Figure 1?" Studying Figure 16 in [9][12] while thinking about this question leads to the hypothesis that a minimal structure would include Loops 1 and 2 delineated in Sections I and II above, along with at least one of either Loop 3: progress → Undiscovered Rework → perceived progress → rework discovery → Known Rework → progress; or, Loop 4: progress → Undiscovered Rework → perceived progress → expected hours at completion → staffing requested → hiring → Staff →equivalent staff on project → progress. Having such a qualitative hypothesis means that, for the oscillatory *pattern-over-time* in Figure 1, we've completed a first pass for the first activity in the *System Structure* step and that we are ready to exercise the other activities in the *System Structure* step.

There is one advantage of the STim process as compared to most other system dynamics modeling processes (e.g., Chapter 3 of [16]). Step 4 asks STim practitioners not only to hypothesize the current mental model creating the problem dynamics, but also to hypothesize a better mental model that will improve the dynamics. The "*Hypothesize improved mental models that could achieve the desired patterns-over-time*" portion of Step 4 of the STim process provides an explicit approach for defining policies that can improve performance over time. For example, in the case of this paper the current mental model creating the problem dynamics is the Productivity Loop in Figure 3; the better mental model that would improve performance is the combined Productivity and Quality Loops in Figure 6. The challenge now is how to shift the mental models of those with the authority to accelerate projects.

One technology that can play a key role in facilitating such learning and personal development is a STim tool called a management flight simulator. Developing such a simulator would involve appropriately embedding the project management simulation in this paper into a computerized learning environment for those who have, or will have, authority to accelerate projects and programs. At present we have the same problem that Senge and Sterman [14] described after a management team at Hanover Insurance had developed new understandings that required adoption by people throughout the company:

"*The problem now facing the team was how to develop shared understanding throughout the organization. The managers who went through the intense learning process* [e.g. the development of Parts I and II of this paper] *could not expect those who had not to agree with its 'counterintuitive' implications. At Hanover, and increasingly in other firms, decision-making responsibility* [e.g. program/project acceleration decision-making authority] *is widely distributed. There are hundreds of individuals who implement new policies and may easily thwart new initiatives. For significantly new policies to come into practice, each person must go through their own personal learning process.*"

V: Conclusion

Parts I and II of this paper use Systems Thinking aided by Simulation (STim) to develop a hypothesis of how projects and programs can finish later and cost more than originally planned despite having been accelerated from an initial feasible schedule for the purpose of ensuring that they finish on time. Computer simulation tests support the hypothesis. The hypothesis is that our mental models take a productivity feedback loop into account when we accelerate projects and programs, but either partially discount or totally ignore a quality feedback loop that counteracts the productivity loop. Part III describes the STim process' continuum of practice, and employs project management examples to illustrate the continuum. The STim work on generic projects/programs in Parts I and II is illustrative of one end of the continuum, whereas much more complex STim work on specific projects/programs is used to illustrate the other end of the continuum. Part IV of this paper describes the STim process used to develop and test this hypothesis. One key feature of the STim process is that it provides an explicit approach for identifying policies that can improve performance over time.

[13] For example, see Figure 1 in [18]

References

[1] Cooper, Ken & David Andersen. (2017). "Commercial Applications of System Dynamics as Oral Histories: The Ken Cooper Series," The System Dynamics Society. Accessed on: Oct. 23, 2018.

[2] Ford, David N. (2007). "A Bibliography of System Dynamics Project Management Work," Wiley Online Library. Accessed on: Oct. 23, 2018.

[3] Forrester, Jay. (1961). *Industrial Dynamics*. Cambridge, MA: MIT Press.

[4] Johnson, David C., George M. Drakeley, Thomas N. Plante, William J. Dalton, & Christopher S. Trost. (2007). "Managing Change on Complex Programs – VIRGINIA Class Cost Reduction." [Paper provided to author by William J. Dalton.]

[5] Lyneis, James M, Kenneth G. Cooper, & Sharon A Els. (2001). "Strategic Management of Complex Projects: a Case Study Using System Dynamics," *System Dynamics Review*, *17(3),* 237-260. Accessed on: Oct. 23, 2018.

[6] Merritt, Jeremy. (2010a). "What are Mental Models?" *The Systems Thinker*, *21(2)*. Accessed on: Oct. 23, 2018.

[7] Merritt, Jeremy. (2010b). "What are 'Mental Models'? Part 2," *The Systems Thinker, 21(9)*. Accessed on: Oct. 23, 2018.

[8] Rahmandad, Hazhir, Rogelio Oliva, & Nathaniel D. Osgood. (2015). *Analytical Methods for Dynamic Modelers*. Cambridge, MA: MIT Press.

[9] Reichelt, Kimberly and James Lyneis. (1999). The Dynamics of Project Performance: Benchmarking the Drivers of Cost and Schedule Overrun, *European Management Journal, 17(2),* 135-150. Accessed on: Oct. 23, 2018.

[10] Repenning, Nelson P. (2000). "A Dynamic Model of Resource Allocation in Multi-project Research and Development Systems," *System Dynamics Review, 16(3),* 173-212.

[11] Richardson, George. (2011). "What is SD? Introduction to System Dynamics," *System Dynamics Society*. Accessed on: Oct. 23, 2018.

[12] Saeed, Khalid. (1992). "Slicing a Complex Problem for System Dynamics Modelling," Proceedings from the 10th International Conference of the System Dynamics Society, Utrecht, Netherlands. Accessed on: Oct. 23, 2018.

[13] Senge, Peter. (1990, 2006). *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY: Doubleday.

[14] Senge, Peter M., & John Sterman. (1992). "Systems Thinking and Organizational Learning: Acting Locally and Thinking Globally in the Organization of the Future," *European Journal of Operational Research*, *59(1)*, 137-150.

[15] Sterman, John D. (1984). "Appropriate Summary Statistics for Evaluating Historical Fit of System Dynamics Models," *Dynamica*, *10*, 51-66. Accessed on: Oct. 23, 2018.

[16] Sterman, John D. (2000). Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin McGraw-Hill.

[17] Sterman, John D., Nelson P. Repenning, & Fred Kofman. (1997). "Unanticipated Side Effects of Successful Quality Programs: Exploring a Paradox of Organizational Improvement," *Management Science*, *43(4)*. Accessed on: Oct. 23, 2018.

[18] Triantis, Kostas, Hazhir Rahmandad, & Warren Vaneman. (2009). "Systems Engineering: Have we Lost our Competitive Edge? A Consideration of the Dynamics of Systems Engineering Projects," Proceedings from the 27th International Conference of the System Dynamics Society, Albuquerque, New Mexico. Accessed on: Oct. 23, 2018.

Appendix 1: Full model sketch and equations

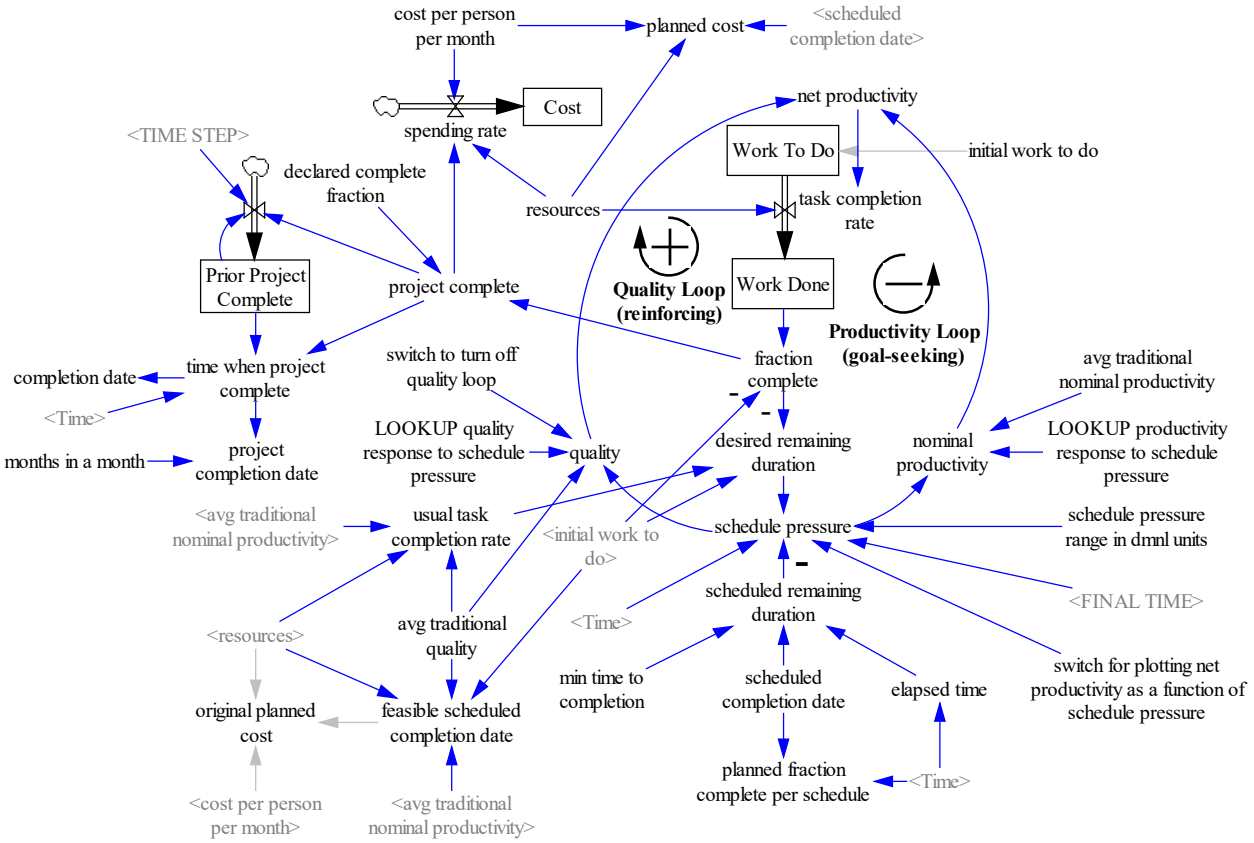


Figure 10: Sketch of complete simulation model that produced the simulation runs in the paper

Levels (or Stocks):

Cost= INTEG (spending rate, 0) Units: $

Prior Project Complete= INTEG ((project complete - Prior Project Complete) / TIME STEP, 0) Units: dmnl

Work To Do= INTEG (-task completion rate, initial work to do) Units: task

Work Done= INTEG (task completion rate, 0) Units: task

Rates (or Flows):

spending rate=resources * cost per person per month * (1 - project complete) Units: $/ Month

task completion rate=resources * net productivity Units: task / Month

Feedback Loop Auxiliaries:

desired remaining duration=(1 - fraction complete) * ZIDZ (initial work to do , usual task completion rate) Units: Month

fraction complete=Work Done / initial work to do Units: fraction

net productivity=nominal productivity * MIN (1 , quality) Units: task / Month / person

nominal productivity=avg traditional nominal productivity * LOOKUP productivity response to schedule pressure (schedule pressure) Units: task / Month / person

quality=avg traditional quality * IF THEN ELSE (switch to turn off quality loop = 0 , LOOKUP quality response to schedule pressure (schedule pressure) , 1) Units: fraction

schedule pressure=IF THEN ELSE (switch for plotting net productivity as a function of schedule pressure = 1 , schedule pressure range in dmnl units * Time / FINAL TIME , ZIDZ (desired remaining duration , scheduled remaining duration)) Units: fraction

Lookup Functions:[14]

LOOKUP productivity response to schedule pressure ([(0,0)-(2,2)],(0,0),(0.3,0.1),(0.45,0.23),(0.6,0.5),(0.8,0.8),(1,1),(1.2,1.13),(1.6,1.3),(2,1.4) Units: fraction

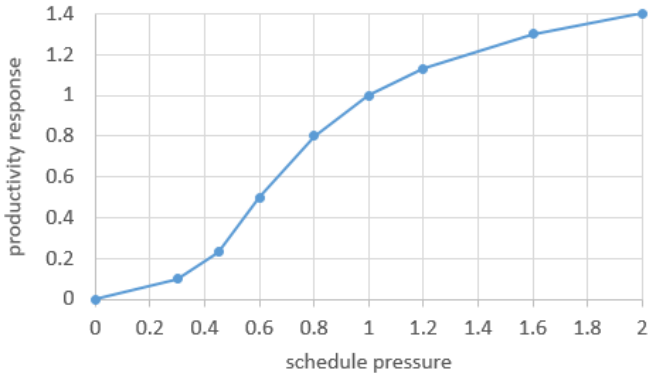


Figure 11: Nonlinear response of productivity to schedule pressure. See discussion in the last two paragraphs of Section II-A.

LOOKUP quality response to schedule pressure ([(0,0.6)-(2,1.4)],(0,1.06),(1,1),(1.1,0.98),(1.2,0.96),(1.3,0.92),(1.4,0.86),(1.5,0.8),(1.6,0.72),(1.7,0.66),(1.8,0.63),(2,0.6)) Units: fraction

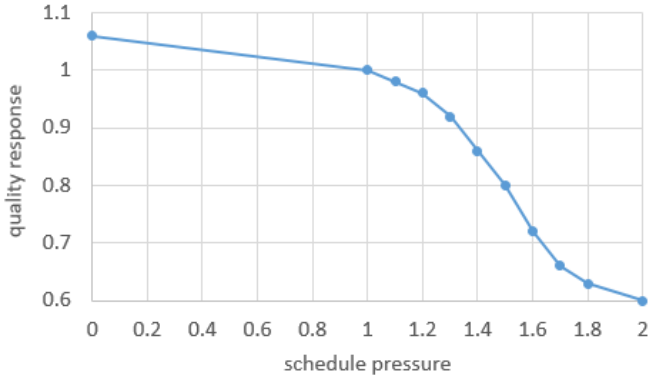


Figure 12: Nonlinear response of quality to schedule pressure. See discussion in the next-to-last paragraph in section II-B.

Auxiliaries that are not part of feedback loops:

completion date=SAMPLE IF TRUE(time when project complete > 0 , time when project complete, time when project complete) Units: Month

elapsed time=Time Units: Month

[14] Lookup functional forms are from Figure 1 of [18].

feasible scheduled completion date=initial work to do / (avg traditional nominal productivity * avg traditional quality) / resources Units: Month

original planned cost= INITIAL(cost per person per month * feasible scheduled completion date * resources) Units: $

planned cost=scheduled completion date * cost per person per month * resources Units: $

planned fraction complete per schedule=IF THEN ELSE (Time <= scheduled completion date , Time / scheduled completion date , 1) Units: fraction

project complete=IF THEN ELSE (fraction complete >= declared complete fraction , 1 , 0) Units: dmnl

project completion date=time when project complete / months in a month Units: fraction

scheduled remaining duration=IF THEN ELSE (scheduled completion date - elapsed time > min time to completion, scheduled completion date - elapsed time, min time to completion) Units: Month

time when project complete=IF THEN ELSE (project complete > Prior Project Complete , Time , 0) Units: Month

usual task completion rate=resources*avg traditional nominal productivity*avg traditional quality Units: task/Month

Parameters:

avg traditional nominal productivity=1 Units: task / Month / person

avg traditional quality=0.9 Units: fraction

cost per person per month=15000 Units: $/ person / Month

declared complete fraction=0.99 Units: fraction

initial work to do=720 Units: task

min time to completion=1 Units: Month

months in a month==1 Units: Month

schedule pressure range in dmnl units=2 Units: dmnl

scheduled completion date=40 Units: Month

switch for plotting net productivity as a function of schedule pressure=0 Units: dmnl
Set this switch to 1 to create a plot with "schedule pressure" on the x axis and "net productivity" on the y-axis.

switch to turn off quality loop=0 Units: dmnl

resources=20 Units: person

Simulation Control Parameters:

INITIAL TIME = 0 Units: month

FINAL TIME = 48 Units: month

SAVEPER =TIME STEP Units: month

TIME STEP = 0.03125 Units: month

Notes on functions used in equations:

INITIAL (A): Returns the value of A at initialization time and holds it constant thereafter. It is used to record and hold or "remember" a variable's starting value.

INTEG (rate, initial value) = Numerical INTEGration. Returns the integral of the rate. The rate is numerically integrated. The initial value is the value of the variable on the left-hand side of the equation at the start of the simulation.

LOOKUP: y=LOOKUP(x) is equivalent to y=f(x). The lookup table is entered with the value of x and returns the y value associated with x in the table or graph. Linear interpolation is used between (x, y) points in the table. If the x input is outside the range of the x values in the table, then LOOKUP returns the y value for the table x value that is closest to the x input.

SAMPLE IF TRUE (condition, input, initial value): SAMPLE input IF condition is TRUE and then hold. Returns input when condition is true and otherwise remains constant. The function initially holds constant at the stated initial value. This function is useful for retaining information about a variable's behavior.

ZIDZ(A,B): Zero (0,0) if dividing by zero (B=0) otherwise returns A divided by B. It is normally used to express the special case where the limit of A/B, as B approaches 0, is 0.

Appendix 2: Instructions for replicating simulation output in figures

Figure 4:

1) Create an *Original Schedule* data set by simulating the model using the settings in Appendix 1.
2) Create an *Actual Schedule* data set by setting:
   a. *switch to turn off quality loop* to a value of 1.
   b. *scheduled completion date* to a value of 30 months as shown in the "Accelerated" column in the table in Figure 4.
3) Plot the variables in Table 1 at right:
4) The data in the table is:
   a. Original schedule performance (months)
      i. Variable: *completion date* @ 48 months
      ii. From dataset: Original Schedule
   b. Accelerated schedule performance (months)
      i. Variable: *scheduled completion date*
      ii. From dataset: Actual Schedule
   c. Simulated schedule performance (months)
      i. Variable*: completion date* @ 48 months
      ii. From dataset: Actual Schedule
   d. Original cost performance (millions of $s)
      i. Variable: *original planned cost*
      ii. From dataset: Original Schedule
   e. Accelerated cost performance (millions of $s)
      i. Variable: *planned cost*
      ii. From dataset: Actual Schedule
   f. Simulated cost performance (millions of $s)
      i. Variable: *Actual Cost* @ 48 months
      ii. From dataset: Actual Schedule

Variable Name	Dataset	Fig 4 Line Color
fraction complete	Original Schedule	sloped red
project complete	Original Schedule	vertical red
planned fraction complete per schedule	Actual Schedule	sloped blue
fraction complete	Actual Schedule	sloped green
project completion date	Actual Schedule	vertical green

Table 1: Variables to plot for Figure 4

Figure 5:
1) Set *switch to turn off quality loop* to a value of 1.
2) Run multiple simulations setting *scheduled completion date* to values of 40, 39, 38…20 months. For each simulation record the value of *completion date* at the end of the simulation. Plot the results: *scheduled completion date* on the x-axis and *completion date* on the y-axis.

Figure 7:
1) Create an *Original Schedule* data set by simulating the model using the settings in Appendix 1.
2) Create an *Actual Schedule* data set by setting *scheduled completion date* to a value of 30 months as shown in the "Accelerated" column in the table in Figure 4.
3) Plot the variables as shown in the table in Appendix 2 - Figure 4 - number 3.
4) The data in the table can be obtained by following the directions in Appendix 2 - Figure 4 - number 4.

Figure 8:
The red graph is the same as the graph in Figure 5.

To produce the green graph, run multiple simulations setting *scheduled completion date* to values of 40, 39, 38…20 months. For each simulation record the value of *completion date* at the end of the simulation. Plot the results: scheduled completion date on the x-axis and completion date on the y-axis.

Appendix 3: Modeling commentary

This appendix contains additional details of potential interest to simulation and modeling practitioners. First, the phrase "Systems Thinking aided by Simulation," or "STim" for short is the same thing as System Dynamics [11]. The reason for use of the STim phrase and acronym is that systems thinking has a significant following, and so using "Systems Thinking aided by Simulation" builds on that understanding rather than causing people to spend a lot of time trying to differentiate between Systems Thinking and System Dynamics.

Second, regarding variations, stochastic or otherwise, of either the model's parameters or its nonlinear functions to capture the full range of dynamics the model can produce, the focus of this work is on a very specific question: can the acceleration of a project/program cause the project/program to finish later than originally planned before the acceleration? Other than the range of variation employed in the *scheduled completion date* parameter as discussed in the text and in Appendix 2, parameter and nonlinear function variation are not necessary to answer this question. The only necessity is that the model's structure, parameters and nonlinear functions be collectively plausible. This is not to say that new insights would not be discovered were the model's parameters and nonlinear functions to be varied across the full range of their plausible values. Indeed, new insights might well be discovered. However, such discoveries were not the objective of this work. Of course STim (System Dynamics) does allow such variable simulations. Indeed, the System Dynamics literature contains many models in which much variability is used, stochastic and otherwise. For examples see especially [8] and [15], as well as [3], [16], and [18]. Note that the publication dates of these references range from 1961 through 2015.

Third, note that the simulation plots in Figure 7 display minimal nonlinear dynamics. This is just one set of dynamics produced by the model, and the output of multiple simulation runs of the model are captured in Figure 8. The model's nonlinear responses are really revealed in Figure 8. The next to last paragraph in Section II-B explains how the two nonlinear feedback loops in the model combine to create these nonlinear results. And the nonlinearities interacting to create these dynamics are shown in Figure 11 and Figure 12. Just as single simulation runs of linear models can produce non-linear dynamics [16, Chapter 8], single simulation runs of nonlinear models don't necessarily produce nonlinear dynamics.

Finally, in [9]'s Figure 16, no hash marks signifying delays are shown across causal links. Many delays are shown in [9]'s Figure 16, but not in the form of hash marks across causal paths, but rather in the form of stocks (state variables). Every feedback loop must contain at least one stock, and every stock involves a delay. So there are delays shown in [9]'s Figure 16. However, there are many causal links in [9]'s Figure 16 that do have delays in them that could be shown by hash marks (they could also be shown using additional stocks). The purpose of [9]'s Figure 16 is to convey the feedback loop complexity, and some of the more salient stock-flow complexity, of a development program. Hash marks were intentionally not shown to avoid complicating the figure. For a discussion of delays in the dynamics of projects and programs see Sections 2.3 and 6.3.4 in [16]. For a thorough treatment of delays in general in dynamic systems, including projects and programs, see Section 5.2.5 and Chapter 11 in [16].